VBScript controls blocked in Microsoft 365 client applications

> **Note:** This article has done its job, and will be retiring soon. To prevent "Page not found" woes, we're removing links we know about. If you've created links to this page, please remove them, and together we'll keep the web connected.

Last updated on August 14th, 2018

For security reasons, new builds of Microsoft Office for Microsoft 365 on Windows block activation of VBScript controls. Most users won't be impacted, but some users may see this runtime error when they try to use an add-in or VBA solution that depends on VBScript:

```
'380' - 'A script engine for the specified language cannot be created.'
```

There are two ways you can unblock these controls. The first is by downloading a small package of registry keys. The second, by manually editing the registry yourself.

Download a package of registry keys

To unblock these controls, click here to download the free package of registry keys. There are easy instructions on the download page to tell you how to proceed.

I'd rather do it myself

To re-enable VBScript controls manually you'll need to modify the Windows registry.

> **Caution:** Follow these steps carefully. Serious problems may occur if you modify the registry incorrectly. Before you start we recommend that you have a known good backup of your registry. See this article for more information: How to back up and restore the registry in Windows..

1. Exit all Microsoft Office applications.

2. Start the Registry Editor by clicking **Start** (or pressing the Windows key on your keyboard) then typing `regedit` and pressing `enter`.

3. Locate the proper registry subkey. It will be either:

   For 64-bit Office or 32-bit Office on 32-bit Windows

   `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Office\16.0\Common\COM Compatibility\`

   or

   For 32-bit Office on 64-bit Windows

   `HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Microsoft\Office\16.0\Common\COM Compatibility\`

   Tip: You can use CTRL+F to quickly find subkeys in the Registry Editor.

   Note: If the **COM Compatibility** key doesn't exist on your machine, you'll need to create it. Right-click the ..\Common node and select **New** > **Key**. Name the new key `COM Compatibility` then proceed with step 4.

4. Add a new subkey to **COM Compatibility** by right-clicking **COM Compatibility** and choosing **New** > **Key**. The new key will be named for the CLSID of the control you want to unblock.

Control	CLSID
VBScript Language	{B54F3741-5B07-11cf-A4B0-00AA004A55E8}
VBScript Language Authoring	{B54F3742-5B07-11cf-A4B0-00AA004A55E8}
VBScript Language Encoding	{B54F3743-5B07-11cf-A4B0-00AA004A55E8}

   Usually you will only need to use the first one in the table (VBScript Language). If you find you need to unblock additional controls, just repeat steps 4-8 for each additional control.

5. Right-click the new key you just created in step 4 and choose **New** > **DWORD (32-bit) value**.

6. Give your new subkey the name of `Compatibility Flags` and the value of `0`.

7. Right-click the new key you created in step 4 and choose **New** > **DWORD (32-bit) value**.

8. Give your new subkey the name of `ActivationFilterOverride` and a value of `1`.

9. Exit the Registry Editor.

The control you need should now be unblocked.